**Introduction to Design**

Design is the process in which user requirements are transformed into some useful forms which further helps a programmer in Coding or Implementation phase. Design is done in software development because it helps in preventing redundancy increases reusability and it is the best approach to mitigate the risk that we have no idea about.

**Structural Diagram**

**Final Class Diagram:** Class diagram in UML (Unified Modelling Language) is a static diagram that describes the system structure by demonstrating the system classes, attributes, operations and relationship that exists among the objects.

**Justification for the approach taken**

For following reasons, I have made use of class diagram:

1. It shows static structure and classifiers in a system.
2. It is useful for developer as well as team member too because this diagram shows different class and object and relationship that exists between them.
3. It also describes the functionalities done by the system.

**Notations Used**

**Class**



**Attribute**



**Generalization**



**Association**



**Multiplicity**



**Aggregation**



**Actual Diagram**

**Description of Diagram**

**Data Flow Diagram:** It is a diagram that models the flow of information for any information system or process. This modelling techniques helps to show the process that are involved in a system to make transfer of data from the input to the database (File Storage) and report production.

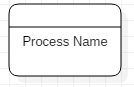
**Justification for the approach taken**

DFD helps in representing process or functions which manipulate, captures, stores and allocate data between the systems. For following reasons, I have used DFD diagram:

1. It helps in describing the logical flow of system.
2. This diagram uses simple notation so it easily understandable.
3. It also helps in creation of automated and manual system requirements.

**Notations Used**

**Process**



**Database**



**Data Flow**



**External Entity**



**Actual Diagram**

**Description of Diagram**

**Behavioral Diagram**

**Activity Diagram:** It is an important diagram that models the dynamic aspects of the system. It is a kind of flowchart that helps to represent the flow from one activity to another activity. The flow can be sequential, concurrent or branched.

**Justification for the approach taken**

For Following reasons, I have made use of activity diagram:

1. It helps in describing the activity flow of system.
2. It describes the sequence among different activity.
3. It defines branched, concurrent, sequential flow of system.

**Notations Used**

**Start Point/ Initial**



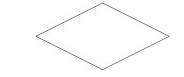
**Activity**



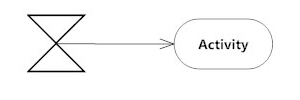
**Action Flow**



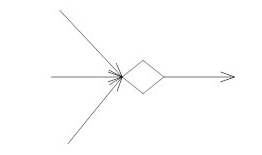
**Decision**



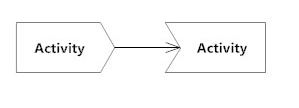
**Time Event**



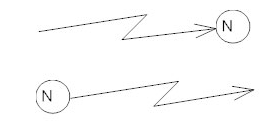
**Merge**



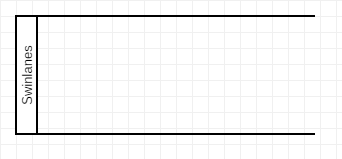
**Message sent and received**



**Interrupting Edge**



**Swim lanes**



**Actual Diagram**

**Description of the Diagram**

**Sequence Diagram:**